**1.0 INTRODUCTION**

This project is to develop to simplify the everyday complicated task performed by the people and securing valuable things. The car shed is built with finest technology used today. The car owns the shed by its automatic reverse system provided by the shed .The shed can be operate with owners smart device even if the owner is in another side of the world.

When unauthorized subjects try to open the shutter without the consent of the owner, the shutter remains locked. The shutter will open by face recognition, number plate recognition. It’s the owner’s choice to choose the subject of shutter. If the face of the subject is not recognized by the help of smart devices, the owner can decide what to do.

The RFID sensor is used for scanning they could present in the vehicle .

This function is performed when the face recognition fails to recognize the subject. The PIR sensor located in the shed performs the function of producing a time delay of closing the shutter while the owner just moved outside.

When the car is been park by the owner the backside of the car faces the shutter, so it is not easy to reverse the car from the inside. That’s why the circular platform automatically rotates 180degree. This makes it easy to move the car outside.

IOT technology is been used in the shed which gives the owner complete control over the shed using the internet

**2.0 PROJECT IN BRIEF**

**2.1 AIM**

To design **SMART SHELTER** using RASPBERRY PI 3 B+ for controlling the working of Shelter

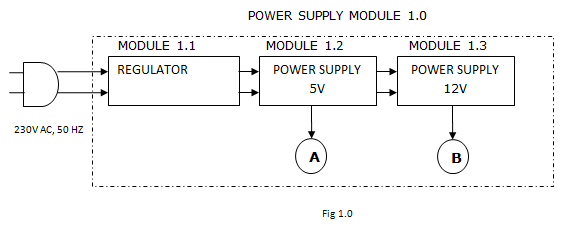
**2.2 OBJECTIVES**

* To have basic ideas about Smart Shelter.
* To understand the working and construction.
* To have a basic idea about working of different sensor.
* Knowledge in selection of components according to the function and requirements.
* To have the basic idea about mechatronics system design.

**3.0 PROJECT TARGET**

* IOT controlled.
* Face recognition.
* RFID recognition.
* Reduce human effort.
* Easy to implement.
* Low maintenance.

**4.0 MODULAR BLOCK DIAGRAM**

****

A

230V AC, 50Hz

RECTIFIER &

FILTER

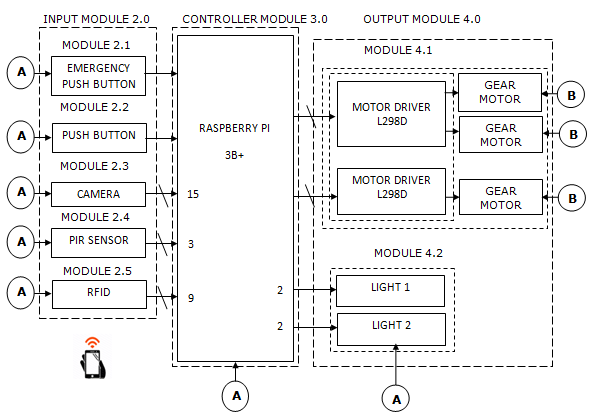
B

POWER SUPPLY

5V, 1A

POWER SUPPLY

12V, 3A

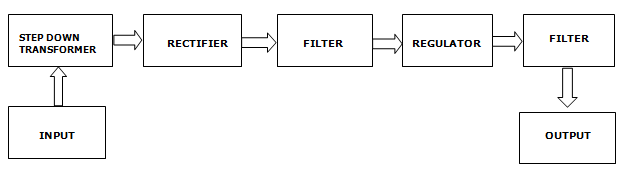
****

SWICTH

Fig 1.1

**5.0 POWER SUPPLY CIRCUIT**

**5.1 BLOCK DIAGRAM**

****

**INPUT**

The input given to the transformer is 230V AC,50Hz

**STEP DOWN TRANSFORMER**

Most of the logic and digital circuit will work properly only in a voltage that is less than 230V. That is why we are using a step down transformer in a power supply circuit.

**BRIDGE WAVE RECTIFIER**

To convert an AC signal into DC signal we are using rectifier circuit, which is build up of rectifier diodes. Output of the rectfier circuit may contain ripples.

**FILTER CIRCUIT**

To remove the ripples in the output we are using fillter circuit which is build up of capacitor

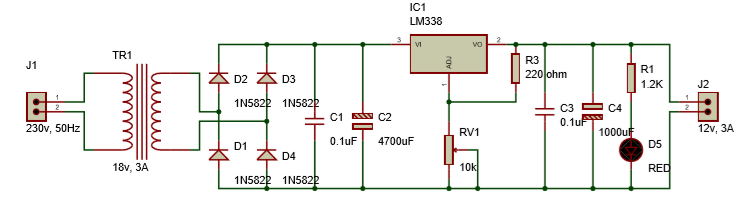
**REGULATOR**

We are using variable voltage regulator ICs to maintain an adjustable constant output voltage in power circuit. We use regulator IC LM338 for getting suitable charging voltage.

**OUTPUT**

The output of the power supply is 12V DC

**5.2 12V, 3A POWER SUPPLY CIRCUIT DIAGRAM**

****

**5.3 12V, 3A CIRCUIT DESIGN**

**SELECTING THE VALUE OF RESISTOR R1**

Resistor is used to limit the current through LED.

Current through the LED is 10mA.

Value of resistor = 12-1.5v/10mA

=1.05kΩ

Value of resistor~1.2kΩ

So selected resistor value is 1.2kΩ

PR1=V\*I

=10.5\*10Ma\*\*

=105mW.

Design margin for PR1=4 times of 105mW

= 420mW.

There for, PR1=1/2W.

So selected resistor, R1=CFR, 1.2kΩ, 1/2W.

**SELECTING INDICATOR LED (D5)**

D5= RED, Ф5mm, 10mA.

**SELECTING THE VALUE OF CAPACITOR (C4)**

Capacitor C4 is introduced because, if there is a failure in supply to the input side, the capacitor C4 can deliver supply to the load for a millisecond of time. For that we should the value of C4 for 1millisecond.

Resistance of load will be = Voutput/Ioutput

That is, R load = 12V/3A.

Load resistance, RL = 4Ω.

We have time constant, RL\*C4=1ms.

Ie, IR\*C4=1\*10^-3

C4=(1\*10^-3)/4.

= 250uF

For safety of the capacitor, we take the 4 time of the value of capacitor.

Ie,4\*C4=4\*250uF=1000uF.

Design margin for a capacitor C4=1000uF.

Selected value of the capacitor, C4=1000uF, 24V.

**SELECTING THE VALUE OF RESISTOR R3, VR1**

Selecting value of R3

Vout=1.25(1+ RV1/R3)

Assuming R3 as 220 ohm

12V = 1.25(1+ RV1/220)

RV1 =2K ohm

So we select 10k pot for selecting approx. value of 2k to reduce the damage for IC and load resistor.

**SELECTING THE VALUE OF CAPACITOR (C3)**

C3 ≈ 0.1uF

Capacitor C3 is used to filter the noise that is produced by the load. We have an equation that, Xc=1/2nF\*c

Xc means the capacitive resistance of capacitor.

For large frequency noise, we should select the low value of C3. So, we select the value of C3 =0.1uF.

**SELECTING THE VALUE OF CAPACITOR (C2)**

Vrip(p-p) =Idc/(2\*f\*C4)……….(1)

Vrip(p-p) = 2√3\*Vrip(rms)………..(2)

Vrip(p-p) = 0.48\*Vdc…………..(3)

Putting equation (3) on equation (2)

Ie, Vrip(p-p) =2√3\*0.48\*Vdc………….(4)

Equating the equation (1) & (4)

2√3\*0.48\*V = Idc/(2\*f\*C4)

There for C2 = Idc/(4\*2√3\*0.48\*Vdc)

Vdc = 15V

Idc = 3A

Frequency, F = 50Hz

C2 = 3/(2\*50\*2√3\*0.48\*12 = 1200uF

Designed value of capacitor, C2 = 1200uF

Selected value of capacitor, C2 =4700uF, 45V.

**SELECTING THE VALUE OF CAPACITOR (C1)**

Selected capacitor C1 is used for filtering the signal coming from the input side. It should be less value capacitor because less value capacitor can filter high frequency spikes and disturbances.

So selected value capacitor, C1 = 0.1uF.

**SELECTING OF TRANSFORMER (TR1)**

We have, Vrms = Vdc\*∏/2√2

Vdc = 12V

Vrms = 12\*∏/2√2 = 18V

Selected transformer TR1 = 18V, 3A.

**SELECTING THE REGULATOR IC FOR 12V (IC1-LM338)**

PIC = (Vin-Vout)\*Iin

PIC = (18-12)V\*(500\*10^-3)

PIC = 3W

Required specification of regulator IC is LM338, 3W, 5A.

So selected IC is LM338, 3W, 3A (p-p)

**SELECTING THE DIODE FOR 12V (D1, D2, D3, D4)**

Required voltage = 12V

Required current =3A

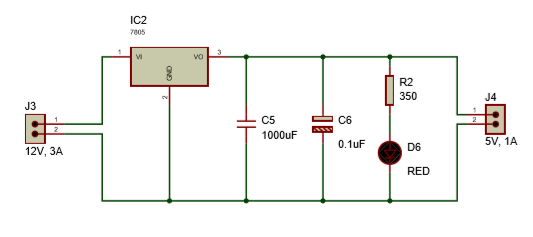
(Average current through each diode are 3A)

D1, D2, D3, D4= 1N5822, 3A, 40V

**5.4 BILL OF MATERIALS FOR 12V POWER SUPPLY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO.** | **ITEM**  **DESCRIPTION** | **SYMBOL** | **QUANTITY**  **(NO’S)** | **PRICE**  **(RS)** |
| 1 | Transformer 18V, 3A | TR1 | 1 | 300 |
| 2 | Diode 1N5822,1/4watt | D1, D2, D3, D4 | 4 | 8 |
| 3 | Capacitor 0.1uf | C1, C3 | 2 | 5 |
| 4 | resistor 1.2k, 220 | R1,R3 | 2 | 8 |
| 5 | Variable resistor | VR1 | 1 | 10 |
| 5 | IC LM338 | IC1 | 1 | 15 |
| 6 | Capacitor 4800uf | C2 | 1 | 5 |
| 7 | Capacitor 1000uf | C4 | 1 | 5 |
| 8 | Connecting wire (multi strand) |  | 1m | 5 |
| 9 | Connecting wire (single strand) |  | 1m | 5 |
|  | |  | TOTAL | **366/-** |

**5.5 5V, 1A POWER SUPPLY UNIT**

****

350Ω

Fig 5.6.1

The output of the power supply is 5V, 1A DC.

**5.6 5V CIRCUIT DESIGN**

**Selecting the Value of Capacitor C6**

C6≈0.1μF (capacitor C6 is used to filter the noises that are produced by load)

We have an equation that XC = 1/2πfc

XC means the capacitive reactance of capacitor C6

For large frequency noises, we should select low value of capacitor C6.So selected value of capacitor C6=0.1μF.

**Selecting the Value of Capacitor, C5**

Capacitor C5 is introduced because, if there is a failure in supply to the input side, the capacitor C5 can deliver supply to the load for a millisecond of time. For that should design the value of capacitor C5 for 1ms.

Resistance of load will be RL=Vout/Iout

RL = 5V/1A

RL = 5KΩ

Assume backup time =1ms

Time constant, RC3 = 1ms

I.e. R x C5 =1 x

Therefore C5 = 1ms/5Ω =200μF

Taking for margin value of capacitor C5 ≈ 4 times of C5 = 1000μF

Therefore selected value of capacitor C5 = 1000μF, 25V

**Selecting the Value of Resistor R2**

Current required for LED=10mA.

Value of R2,

V/I = 5-1.5V/10mA

= 350Ω

P = VI

= 3.5V\*10mA, 3.5V is taken because 1.2V drop is there across LED2.

= 0.35mW

Power rating should be 4 times greater so P=0.35mW\*4

P≈ 0.05W

Standard value = 1/4W.

Selected value of resistor, R2 = CFR, 350Ω, 1/4W.

**Selecting the Regulator IC for 5V (IC2-LM7805)**

PIC = (Vin-Vout) x Iin

PIC = (18-12) v x (500 x 10-3) = 6 x (500 x 10-3)

PIC = 3 W

Required specification of regulator IC is LM7805, TO 220, 3W, 1A

(P-P)

Available regulator IC is LM7805, 3W, 1A (p-p)

So selected IC is LM7805, TO 220, 3W, 1A (p-p)

**5.7 BILL OF MATERIALS FOR 5V POWER SUPPLY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **SL.NO.** | **ITEM**  **DESCRIPTION** | **SYMBOL** | **QUANTITY**  **(N0’S)** | **PRICE**  **(RS)** |
| 1 | Led | D6 | 1 | 5 |
| 2 | Capacitor 1000uf | C5 | 1 | 11 |
| 3 | Capacitor 0.1uf | C6 | 1 | 5 |
| 4 | IC LM7805 | IC2 | 1 | 15 |
| 5 | Resistor 350Ω | R2 | 1 | 10 |
| 6 | Connecting wire (double strand) |  | 1M | 5 |
| 7 | Connecting wire (single strand) |  | 1M | 5 |
| TOTAL | |  |  | **56/-** |

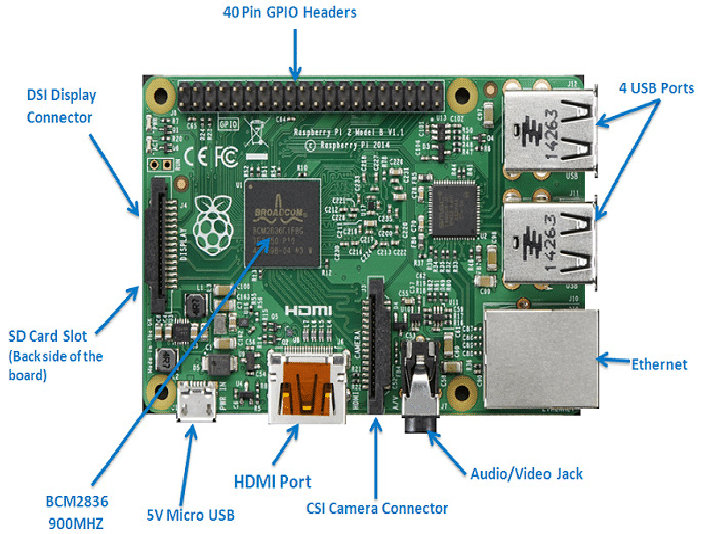
**6.0 CONTROLLER MODULE**

**6.1 RASPBERRY PI**

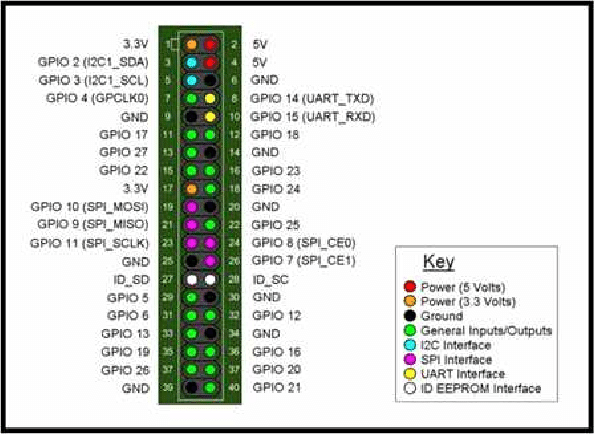
The Raspberry Pi is a small computer that can do lots of things. You plug it into a monitor and attach a keyboard and mouse

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range.

* Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
* 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
* Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
* CSI camera port for connecting a Raspberry Pi camera
* DSI display port for connecting a Raspberry Pi touchscreen display
* 4-pole stereo output and composite video port
* Power-over-Ethernet (PoE) support (requires separate PoE HAT)

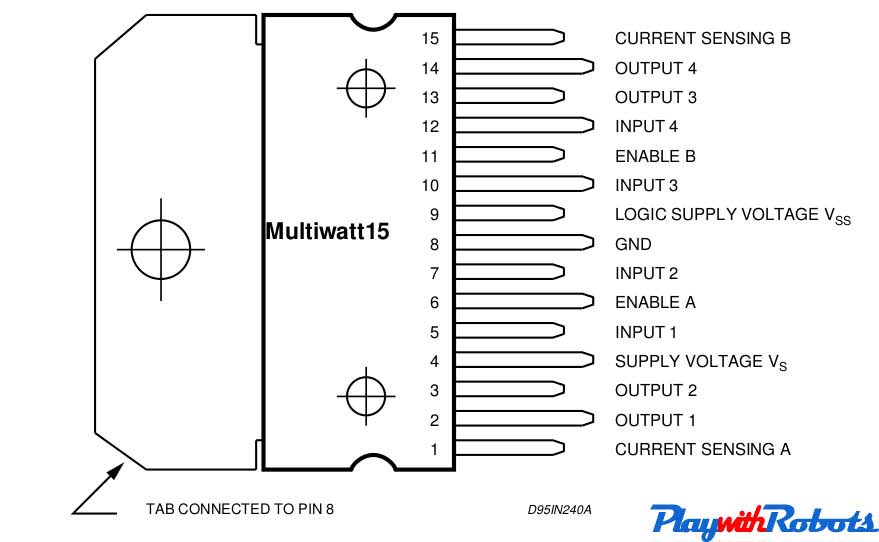
****

**6.2 PIN DIAGRAM**

****

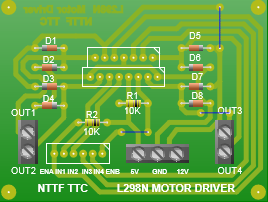
**6.3 ADVANTAGES AND DISADVANTAGES**

|  |  |
| --- | --- |
| **ADVANTAGES**   * Low cost * Small in size * The product also gives you a lot of room to experiment and turn it into something else that is entirely different.  The SD cards on the board can be easily switched, which allows you to change the functions of the device without spending a lot of time re-installing the software. * The product is energy efficient and provides a greener ethical alternative to small businesses. This small credit card sized product makes it easy to recycle and does not release as much carbon dioxide emissions into the environment, unlike big servers that require lots of energy and extensive cooling systems. * The Raspberry Pi is perfect for adaptive technology.   **DISADVANTAGES**   * Not able to run X86 operating system * It does not replace your computer, since the Ethernet is only a 10/100 and the processor is not as fast, it is time consuming to download and install software and is unable to do any complex multitasking. * Not compatible with other operating systems such as Windows There are currently 1.3 billion Windows users around the world (Sawyers, 2012). * To use the Raspberry Pi,it will take more than just 35 dollars to get it to do what you need through buying extra accessories such as the SD card, USB power supply, keyboard etc .   **7.0 MOTOR DRIVER MODULE**  **7.1 DESCRIPTION**  Here we are using two motor drivers. The L298D is an integrated monolithic circuit in a 15-lead Multi watt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive the inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and corresponding external terminal can be used for the connection of external sensing resistor .An additional supply input is provided so that the logic works at a lower voltage.  **7.2 PIN CONFIGURATION** |  |



**7.3 MOTOR DRIVE MODULE FIGURE**

FIG 7.2.1

****

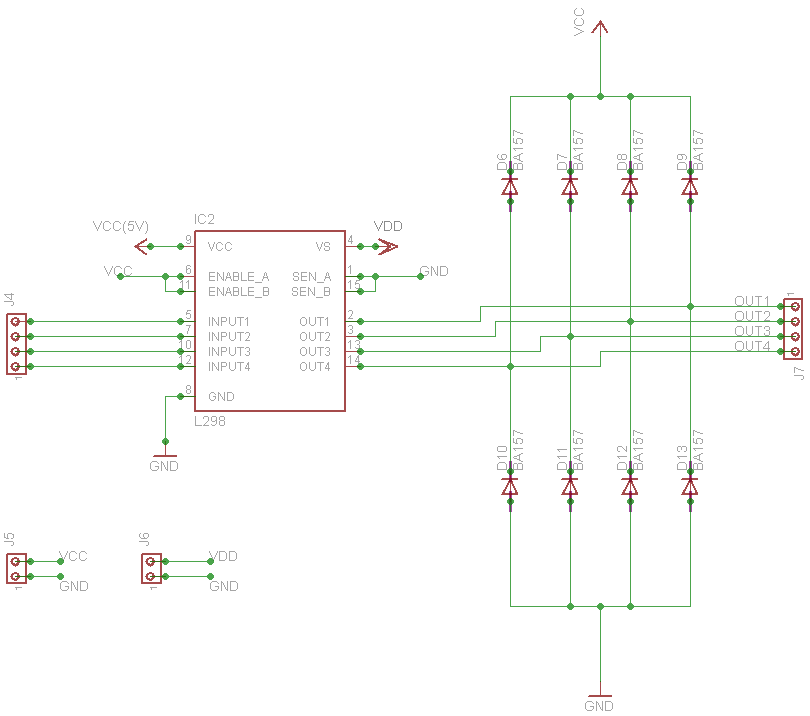
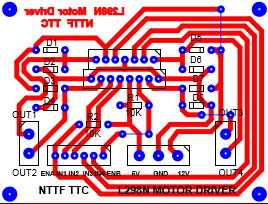
**7.4 CIRCUIT DIAGRAM**

FIG 7.4.1

**7.5 PCB LAYOUT OF MOTOR DRIVER**

****

**7.6 BILL OF MATERIALS FOR MOTOR DRIVER MODULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO** | **ITEM**  **DESCRIPTION** | **SYMBOL** | **QUANTITY**  **(NO’S)** | **PRICE**  **(RS)** |
| 1 | Motor Driver IC L298DMULTIWATT15, DUAL H BRIDGE DRIVER | IC1 | 3 | 210 |
| 2 | DIODE - BY299,2A | D1**,**D2,D3,D4,D5,D6,D7,D8 | 8 | 25 |
| 3 | RELIMATE CONNECTOR - 2 PIN | SL1 | 1 | 36 |
| 4 | RELIMATE CONNECTOR -6 PIN | SL2 | 1 | 30 |
| 5 | RESISTOR - CFR,10K1/4W | R11,R12 | 2 | 25 |
| 6 | CONNECTING WIRE - 0.5MM SINGLE STRAND |  | 1m | 20 |
| 7 | PTB CONNECTOR | X3,X4,X5 | 3 | 30 |
|  | **TOTAL** | | | **376/-** |

**8.0 INPUT MODULE**

**8.1 PIR SENSOR**

****

FIGURE 8.1.1

* Product Type: HC--SR501 body sensor module
* Operating Voltage Range: DC 5V
* Quiescent Current: <50uA
* Level Output: High 3.3 V or low 0V
* HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

**8.2 WEBCAM**

****

FIGURE 8.2.1

This 5mp camera module is capable of 1080p video and still images and connects directly to your Raspberry Pi. Connect the included ribbon cable to the USB port on your Raspberry Pi, boot up the latest version of Rasping and you are good to go!

* Dimensions: 25mm x 20mm x 9mm
* 5 MP resolution
* 2592 x 1944 pixel static images
* 1080p30 video
* This module is only capable of taking pictures and video, not sound.
* 5V,50mA

**8.3 SWITCH**

****

FIGURE 8.3.1

The power button is a round or square button that powers an electronic device on and off. Nearly all electronic devices have power buttons or power switches. Typically, the device powers on when the button is pressed and powers off when the button is pressed again.

**8.4 EMERGENCY STOP BUTTON**

A kill switch, also known as an emergency stop or e-stop, is a safety mechanism used to shut-off a device or machinery in an emergency situation in which it cannot be shut-down in the usual manner. Unlike a normal shut-down switch/procedure, which shuts down all system in an orderly fashion and it turns the machine off without damaging it, a kill switch is designed and configured to completely and as quickly as possible about the operation(even if this damage equipment and be operable in a manner that is quick, simple(so that even a panicking operator with impaired executive function or bystander can activate it).Kill switches are usually designed obvious even to an untrained operator or a bystander



FIGURE 8.4.1

**8.5 RFID**

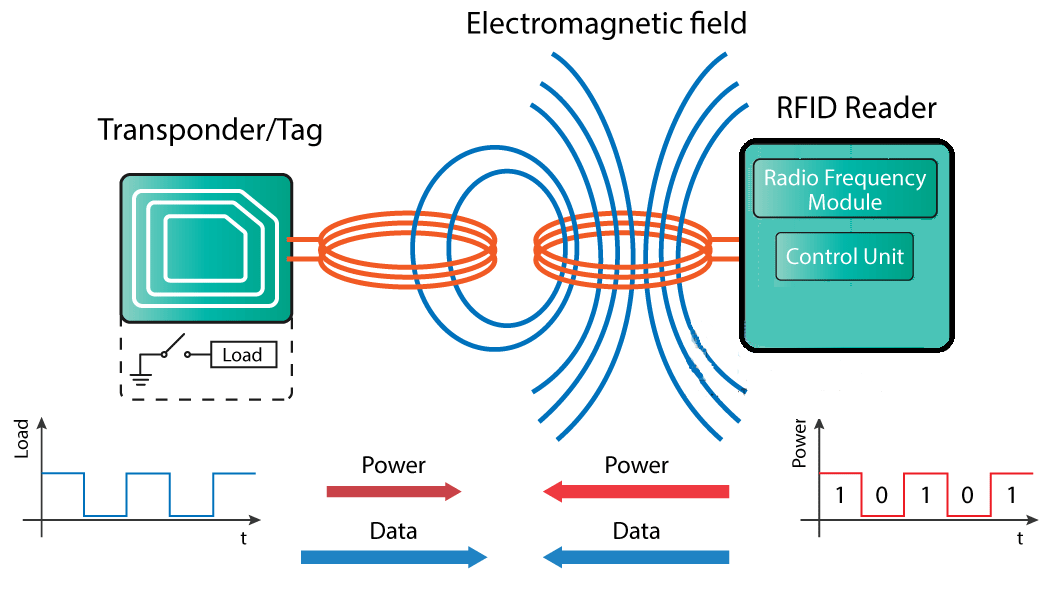
****

FIGURE 8.5.1

RFID is technology which works on radio frequency and it is used for the auto-identification for the different object. The RFID system mainly consists of two parts. ... If the object, on which this RFID tag is attached is within the range of this radio waves then it sends the feedback back to this RFID reader.

**8.6 BILL OF MATERIAL FOR INPUT MODULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **SL.NO** | **ITEM**  **DESCRIPTION** | **SYMBOL** | **QUANTITY**  **(NO’S)** | **PRICE**  **(RS)** |
| 1 | PIR SENSOR |  | 1 | 175 |
| 2 | RFID |  | 1 | 499 |
| 3 | SWITCH |  | 1 | 103 |
| 4 | Emergency Stop button (Mushroom) |  | 1 | 250 |
| 5 | WEBCAM |  | 1 | 424 |
|  | | | TOTAL | **1451/-** |

**9.0 OUTPUT MODULE**

**9.2 GEAR MOTORS**

Gear motors are electric motors that utilize a type of gear system on the output of the motor. This gearing arrangement is called a gear reducer or gearbox. The combination of an electric motor and gearbox reduces design complexity and lowers cost, particularly for motors built for high torque and low speed applications. In addition, gearboxes can be used as a means to reorient the output shaft in a different direction.

****

FIG 9.2.1

**9.3 LED**

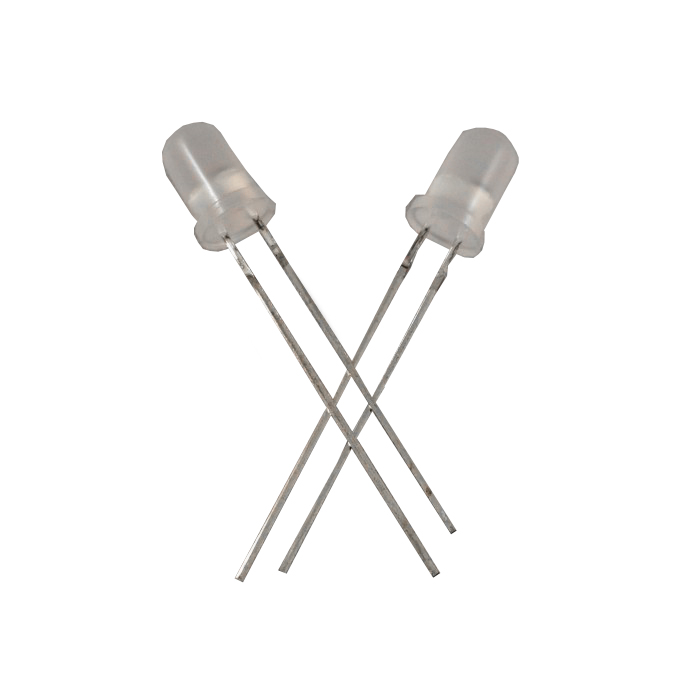
****

FIGURE 9.3.1

* Uniform light output.
* Low power consumption.
* I.C. compatible.
* Long life solders ability.
* Common Cathode.

A light-emitting diode (LED) is a semiconductor device that emits visible light when an electric current passes through it. The light is not particularly bright, but in most LEDs it is monochromatic, occurring at a single wavelength.

**9.5 BILL OF MATERIAL FOR OUTPUT MODULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BILL OF MATERIAL FOR OUTPUT MODULE** | | | | |
| **SL.NO.** | **ITEM DESCRIPTION** | **SYMBOL** | **QUANTITY**  **(NO’S)** | **PRICE**  **(RS)** |
| 1 | L293D |  | 2 | 400 |
| 2 | Gear MOTOR |  | 2 | 939 |
| 3 | LED 4pin |  | 2 | 10 |
|  |  |  | TOTAL | **1349/-** |

**10.0 MECHANICAL STRUCTURE**

**10.1 MECHANICAL DESIGN**

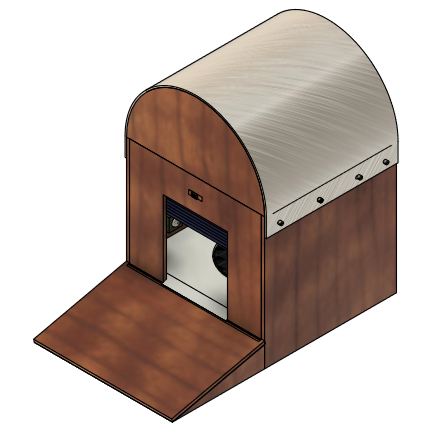


Fig 10.1.1

**10.2 CAD DESIGN**

**FRONT VIEW**

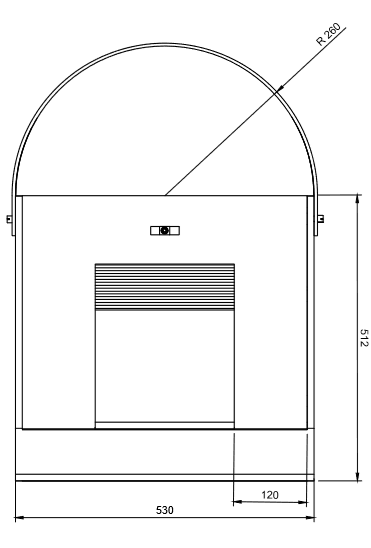


Fig 10.2.1

All dimensions are in mm

**SIDE VIEW**

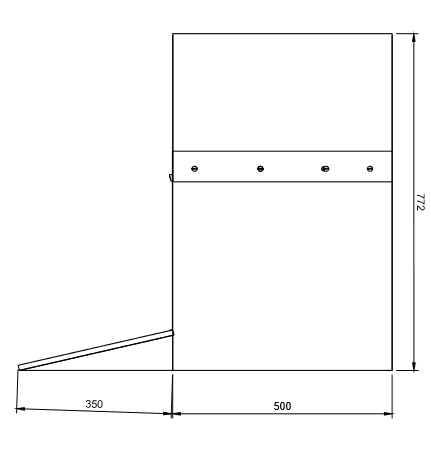


Fig 10.2.2

All dimensions are in mm

**TOP VIEW**

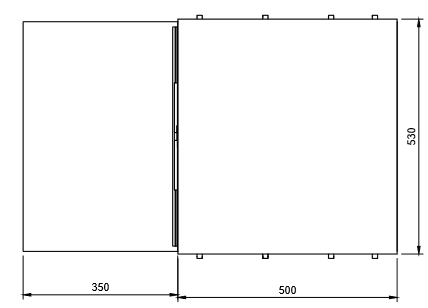


Fig 10.2.3

All dimensions are in mm

**10.3 BILL OF MATERIAL FOR MECHANICAL MODULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BILL OF MATERIAL FOR MECHANICAL STRUCTURE** | | | | |
| **SL.NO.** | **ITEM SPECIFICATION** | **SYMBOL** | **QUANTITY**  **(METRE)** | **PRICE**  **(RS)** |
| 1 | PLYWOOD 1600\*1600\*10mm |  | 5 | 1600 |
| 2 | RUBBER STICKER |  | 1 | 399 |
| 3 | ALUMINIUM SHEET |  | 1.5 | 300 |
| 4 | LETTER BLOCK |  | 12 | 100 |
| 5 | L CLAMP |  | 30 | 100 |
| 6 | PLASTIC SHEET |  | 0.5 | 50 |
|  |  |  | TOTAL | **2249/-** |

**11.0 OVERALL BILL OF MATERIAL OF PROJECT**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **MODULES** | **PRICE(RS)** |
| 1 | POWER SUPPLY MODULE 12V | 366 |
| 2 | POWER SUPPLY MODULE 5V | 56 |
| 3 | CONTROLLER MODULE | 3139 |
| 4 | MOTOR DRIVER MODULE | 376 |
| 5 | INPUT MODULE | 1451 |
| 6 | OUTPUT MODULE | 1349 |
| 7 | MECHANICAL STRUCTURE | 2249 |
|  | **TOTAL** | **8986/-** |

**12.0 SUB ASSEMBLY AND TESTING**

**12.1 POWER SUPPLY UNIT**

In this project we requires one 5V 1A circuit and 12V 3A circuit

**TESTING STEPS**

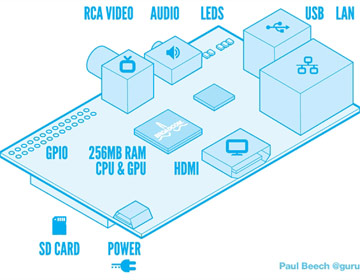
* Checked all components.
* Connected on the PCB board according to the design.
* Gave input (230V AC).
* Verified the output by observing the reading on the multi-meter.
* Soldered the components on a general purpose circuit board.
* Checked for shorts on the circuits by using a multimeter.
* Gave supply and checked whether corrected voltage is coming on the correct pins.

**12.2 CONTROL MODULE**

We are using RASPBERRY Pi as intelligent brain of our system.

The units consist of

* ARM CPU/GPU
* GPIO
* RCA
* AUDIO OUT
* USB port
* HDMI port
* Ethernet port
* 5V micro USB Power supply
* SD card slot
* Wi-Fi connectivity



**TESTING STEPS**

* Checked all the components.
* Connected on the PCB according to the design.
* Gave input (0/1) to pins according to the condition for led blinking circuit.
* 26 GPIO pins can be programmed; these pins go from GPIO2 to GPIO27.
* Connected the circuit as per design.
* Verified the output by switch in the led blinking circuit.
* Gave supply and checked whether corrected voltage is coming on the correct pins.

**12.3 MOTOR DRIVER MODULE**

* We require 2 motor driving units. (L298D)
* It receives input from raspberry pi.
* It drives the gear motor according to the receiving command and programs.

**TESTING STEPS**

* Checked all the components.
* Connected on the PCB according to the design.
* Gave input (0/1) to the pins according to the condition.
* Verified the output by observing the rotation direction of motor.
* Check for shorts on the circuit by using the multimeter.
* Gave supply and checked whether corrected voltage is coming on the correct pin.
* Verify the connection is correct or not.
* Checked the mechanical structure for assembly.
* Gave input and verified the output

**12.4 INPUT MODULE**

* It consists of Pi camera, PIR sensor, emergency stop button, RFID, push button.
* Push button is used to switch on the supply.
* Emergency stop button helps to stop the entire system.
* PIR sensor is used to detecting the motion.
* Her we are using RFID for identify our car.

**TESTING STEPS**

PI camera

* Pi camera consists of a ribbon cable which is used to connect it with the CSI port on Raspberry Pi.
* The Raspistill program is loaded to the python which is used to capture still image.
* Gave supply and runned the program in Raspberry and verified the output. +

PIR sensor

* While we give the supply, the required indication will indicate, whenever a human present senses

RFID

* RFID is a technology which is working on radio frequency of radio waves.
* whenever this object is in the range of the reader then this RFID tag used to transmit its feedback signal to the reader

**12.5 OUTPUT MODULE**

* It consists of speakers, gear motors, LEDs.
* Gear motors are used to rack and pinion and rotating platform.
* LEDs for indicating.
* Speakers is used for giving the command

**TESTING STEPS**

* Give input (0/1) according to the conditions.
* Verified the output by observing the motion of motors.
* Gave supply and checked whether correct voltage is coming on the correct pins.

**13.0 OVERALL ASSEMBLY, TESTING AND OBSERVATION**

**13.1 MECHANICAL**

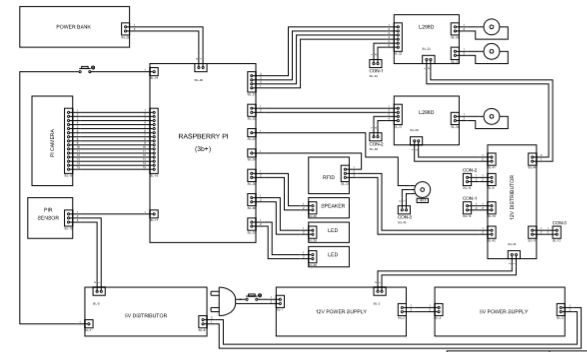
* According to our design made all mechanical parts for our project. The required parts for our project are
* Rotating platform
* Shelter mechanism
* Assembled all mechanical parts

**13.2 ELECTRICAL AND ELECTRONICS**

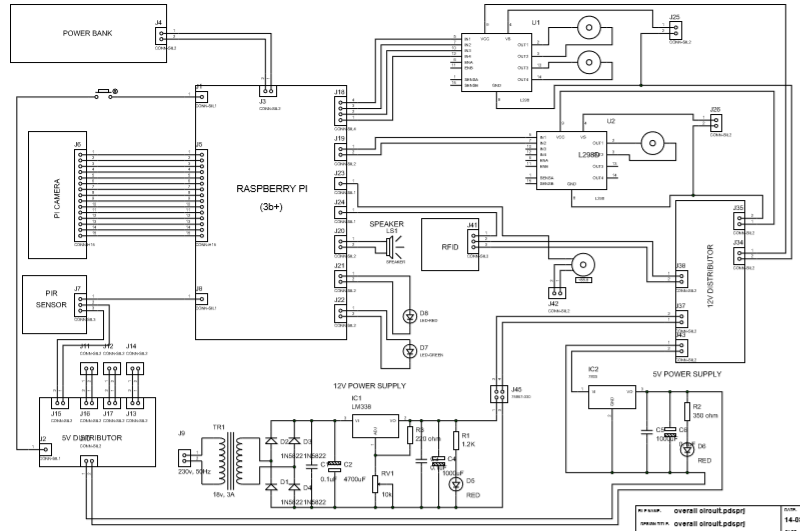
According to our design of each sub circuits made all modules for our project. The required modules for our project are...

* Power supply circuit
* Microcontroller Unit
* Motor Driver Circuit
* From the output of the +5V power supply, made each connection to Microcontroller unit.
* From the output of the +12V power supply, made each connection to Driver circuit.
* The output of the switch circuit and RF Module is connected to the input of the Microcontroller unit.
* From the output of the Microcontroller unit a connection is made to the input of Driver circuit.
* The shaft of the motor then coupled with rack and pinion.
* Checked the continuity of all connections.
* Corresponding switches is pressed and checked the output status.
* Measured the voltage at each input and output terminal

**14.0 INTER CONNECTION DIAGRAM**



**15.0 0VERALL CIRCUIT DIAGRAM**

****

**16.0 FLOW CHART**

MAIL

SEND

IF

EMG.B

=≠

STOP

WAIT 20sec AND CLOSE SHUTER AND OFF

OPEN SHUTER AND TURN ON LIGHT

CLOSE

IF RFID

DECTACTED

IF IOT

ON=1

IF FACE DECTACTED

START

**17.0 PROGRAM**

**MAIN PROGRAM**

import os

def iot():

print("iot START")

o=os.popen(" gnome-terminal -x python3 iotscan.py").read()

print(o)

print("iot STOP")

def rf():

print("rfid START")

o=os.popen(" gnome-terminal -x sudo python3 rfid.py").read()

print(o)

print("rfid STOP")

def face\_de():

print("face START")

o=os.popen(" gnome-terminal -x python3 face.py").read()

print(o)

print("face STOP")

def em():

print("em START")

o=os.popen(" gnome-terminal -x sudo python3 EMER.py").read()

print(o)

if \_\_name\_\_ == '\_\_main\_\_':

try:

iot()

rf()

face\_de()

em()

# Reset by pressing CTRL + C

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**FACE RECOGNITION**

import face\_recognition

import cv2

import numpy as np

import os

video\_capture = cv2.VideoCapture(0)

mahath\_image = face\_recognition.load\_image\_file("mahath.jpg")

mahath\_face\_encoding = face\_recognition.face\_encodings(mahath\_image)[0]

# Create arrays of known face encodings and their names

known\_face\_encodings = [

mahath\_face\_encoding

]

known\_face\_names = [

"MAHATH"

]

# Initialize some variables

face\_locations = []

face\_encodings = []

face\_names = []

process\_this\_frame = True

while True:

# Grab a single frame of video

ret, frame = video\_capture.read()

# Resize frame of video to 1/4 size for faster face recognition processing

small\_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

# Convert the image from BGR color (which OpenCV uses) to RGB color (which face\_recognition uses)

rgb\_small\_frame = small\_frame[:, :, ::-1]

# Only process every other frame of video to save time

if process\_this\_frame:

# Find all the faces and face encodings in the current frame of video

face\_locations = face\_recognition.face\_locations(rgb\_small\_frame)

face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)

face\_names = []

for face\_encoding in face\_encodings:

# See if the face is a match for the known face(s)

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

name = "Unknown"

cv2.imwrite('unKnown0.png',frame)

o=os.popen(" gnome-terminal -x python3 mail.py").read()

print(o)

print("can't open")

#mail.mails()

# # If a match was found in known\_face\_encodings, just use the first one.

# if True in matches:

# first\_match\_index = matches.index(True)

# name = known\_face\_names[first\_match\_index]

# Or instead, use the known face with the smallest distance to the new face

face\_distances = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)

best\_match\_index = np.argmin(face\_distances)

if matches[best\_match\_index]:

name = known\_face\_names[best\_match\_index]

o=os.popen(" gnome-terminal -x python3 open.py").read()

print(o)

print("open")

face\_names.append(name)

process\_this\_frame = not process\_this\_frame

# Display the results

for (top, right, bottom, left), name in zip(face\_locations, face\_names):

# Scale back up face locations since the frame we detected in was scaled to 1/4 size

top \*= 4

right \*= 4

bottom \*= 4

left \*= 4

# Draw a box around the face

cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 255), 2)

# Draw a label with a name below the face

cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)

font = cv2.FONT\_HERSHEY\_DUPLEX

cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

# Display the resulting image

cv2.imshow('Video', frame)

# Hit 'q' on the keyboard to quit!

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release handle to the webcam

video\_capture.release()

cv2.destroyAllWindows()

**RFID**

import serial

import os

import RPi.GPIO as gpio

import time

from time import sleep

def init():

gpio.setmode(gpio.BCM)

gpio.setup(13, gpio.OUT)

gpio.setup(19, gpio.OUT)

gpio.setup(26, gpio.OUT)

def open(sec):

init()

gpio.output(13, False)

gpio.output(19, True)

print("openig")

sleep(sec)

gpio.cleanup()

def LEDON():

init()

gpio.output(26, True)

print("done LED")

gpio.cleanup()

def LEDOFF():

init()

gpio.output(26, False)

print("done LEDOFF")

gpio.cleanup()

def checkc():

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(3, gpio.IN)

i=gpio.input(3)

if i == 1:

open(.1)

#LEDON()

gpio.cleanup()

print("opening",i)

elif i == 0:

break

if \_\_name\_\_ == '\_\_main\_\_':

try:

ser = serial.Serial('/dev/ttyS0', 9600, timeout=1)

while True:

string = ser.read(12)

if len(string) == 0:

print("Please wave a tag")

else:

string = string[1:11] # Strip header/trailer

print(string)

if string == b'0001705948':

checkc()

print("OPEN")

# Reset by pressing CTRL + C

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**OPEN**

import RPi.GPIO as gpio

import time

from time import sleep

import firebase\_admin

from firebase\_admin import credentials

from firebase\_admin import firestore

#servi

cred = credentials.Certificate('/home/pi/PRO/serviceAccount.json')

firebase\_admin.initialize\_app(cred)

db = firestore.client()

def init():

gpio.setmode(gpio.BCM)

gpio.setup(13, gpio.OUT)

gpio.setup(19, gpio.OUT)

gpio.setup(26, gpio.OUT)

def open(sec):

init()

gpio.output(13, False)

gpio.output(19, True)

sleep(sec)

print("openig")

gpio.cleanup()

def LEDON():

init()

gpio.output(26, True)

print("done LED")

gpio.cleanup()

def LEDOFF():

init()

gpio.output(26, False)

print("done LEDOFF")

gpio.cleanup()

def firesetopen():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'Shutter')

doc\_ref.set({

u'Shutter\_pos': u'open'

})

def fireseton():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'LIGHT')

doc\_ref.set({

u'led\_s': u'ON'

})

if \_\_name\_\_ == '\_\_main\_\_':

try:

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(3, gpio.IN)

i=gpio.input(3)

if i == 1:

open(.1)

LEDON()

gpio.cleanup()

print("opening",i)

elif i == 0:

firesetopen()

fireseton()

gpio.cleanup()

print("opened",i)

break

sleep(20)

o=os.popen(" gnome-terminal -x python3 close.py").read()

print(o)

# Reset by pressing CTRL + C

except KeyboardInterrupt:

print("KeyboardInterrupt by User")

gpio.cleanup()

**CLOSE**

import RPi.GPIO as gpio

import time

from time import sleep

import firebase\_admin

from firebase\_admin import credentials

from firebase\_admin import firestore

#servi

cred = credentials.Certificate('/home/pi/PRO/serviceAccount.json')

firebase\_admin.initialize\_app(cred)

db = firestore.client()

def init():

gpio.setmode(gpio.BCM)

gpio.setup(13, gpio.OUT)

gpio.setup(19, gpio.OUT)

gpio.setup(26, gpio.OUT)

def close(sec):

init()

gpio.output(13, True)

gpio.output(19, False)

sleep(sec)

print("closing")

gpio.cleanup()

def LEDOFF():

init()

gpio.output(26, False)

print("done LEDOFF")

gpio.cleanup()

def firesetclose():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'Shutter')

doc\_ref.set({

u'Shutter\_pos': u'close'

})

def firesetoff():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'LIGHT')

doc\_ref.set({

u'led\_s': u'OFF'

})

if \_\_name\_\_ == '\_\_main\_\_':

try:

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(2, gpio.IN)

i=gpio.input(2)

if i == 1: #When output from motion sensor is LOW

#close(4.25)

close(.1)# Reset by pressing CTRL + C

LEDOFF()

gpio.cleanup()

print("closing",i)

elif i == 0:

firesetclose()

firesetoff()

gpio.cleanup()

print("closed",i)

break

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**MAIL**

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from email.mime.base import MIMEBase

from email import encoders

import os

def mails():

#frame = cv2.VideoCapture(0)

#for i in range(2):

# cv2.imwrite('unKnown'+str(i)+'.png', image)

#del(frame)

#o=os.popen('scrot /home/pi/PRO/unKnown0.png').read()

#print(o)

email\_user = 'mmu03091997@gmail.com'

email\_password = 'Qwertyuiop@1'

email\_send = 'mahathmundakkal@gmail.com'

subject = 'subject'

msg = MIMEMultipart()

msg['From'] = email\_user

msg['To'] = email\_send

msg['Subject'] = subject

body = 'AN UNKNOWN FACE DECTECT'

msg.attach(MIMEText(body,'plain'))

filename='unKnown0.png'

attachment =open(filename,'rb')

part = MIMEBase('application','octet-stream')

part.set\_payload((attachment).read())

encoders.encode\_base64(part)

part.add\_header('Content-Disposition',"attachment; filename= "+filename)

msg.attach(part)

text = msg.as\_string()

server = smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

server.login(email\_user,email\_password)

server.sendmail(email\_user,email\_send,text)

server.quit()

if \_\_name\_\_ == '\_\_main\_\_':

try:

mails()

# Reset by pressing CTRL + C

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**IOT SCAN**

import os

import RPi.GPIO as gpio

import time

from time import sleep

def init():

gpio.setmode(gpio.BCM)

gpio.setup(13, gpio.OUT)

gpio.setup(19, gpio.OUT)

gpio.setup(26, gpio.OUT)

def open(sec):

init()

gpio.output(13, False)

gpio.output(19, True)

sleep(sec)

print("openig")

gpio.cleanup()

def close(sec):

init()

gpio.output(13, True)

gpio.output(19, False)

sleep(sec)

print("closing")

gpio.cleanup()

def value():

ch=0

def checkC():

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(2, gpio.IN)

i=gpio.input(2)

if i == 1: #When output from motion sensor is LOW

LEDOFF()

print("closing",i)

elif i == 0:

firesetclose()

firesetoff()

gpio.cleanup()

print("closed",i)

break

def checkO():

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(3, gpio.IN)

i=gpio.input(3)

if i == 1: #When output from motion sensor is LOW

open(1)

LEDON()

print("opening",i)

elif i == 0:

firesetopen()

fireseton()

gpio.cleanup()

print("opened",i)

break

def LIcall():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'LIGHT')

try:

doc = doc\_ref.get()

LED=doc.to\_dict()

STATUS={'led\_s': 'ON'}

if LED == STATUS:

print("ON")

LEDON()

elif LED != STATUS:

print("OFF")

LEDOFF()

except google.cloud.exceptions.NotFound:

print(u'No such document!')

def FRcall():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'FLATFROM')

try:

doc = doc\_ref.get()

FLATFROM=doc.to\_dict()

STATUS={'flatform\_value': 'STOP'}

if FLATFROM == STATUS:

print("STOPED")

elif FLATFROM != STATUS:

print("ROTATE")

o=os.popen(" gnome-terminal -x python3 rotate.py").read()

print(o)

except google.cloud.exceptions.NotFound:

print(u'No such document!')

def SHcall():

value()

ch=0

doc\_ref= db.collection(u'SMART-SHELTER').document(u'Shutter')

try:

doc = doc\_ref.get()

SHUTTER=doc.to\_dict()

STATUS={'Shutter\_pos': 'open'}

if SHUTTER == STATUS:

print("OPEN")

if ch == 1:

checkO()

ch=0

elif SHUTTER != STATUS:

print("CLOSE")

if ch == 0:

checkC()

ch=1

except google.cloud.exceptions.NotFound:

print(u'No such document!')

def iotcall():

doc\_ref= db.collection(u'SMART-SHELTER').document(u'run')

try:

doc = doc\_ref.get()

SHUTTER=doc.to\_dict()

STATUS={'iot': 'on'}

if SHUTTER == STATUS:

SHcall()

FRcall()

LIcall()

elif SHUTTER != STATUS:

print("iotoff")

except google.cloud.exceptions.NotFound:

print(u'No such document!')

if \_\_name\_\_ == '\_\_main\_\_':

try:

while True:

iotcall()

# Reset by pressing CTRL + C

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**LIGHT ON**

From LED\_SETUP import \*

LEDON()

fireseton()

**LIGHT OFF**

from LED\_SETUP import \*

LEDOFF()

firesetoff()

**LED SETUP**

import RPi.GPIO as gpio

import time

from time import sleep

import firebase\_admin

from firebase\_admin import credentials

from firebase\_admin import firestore

import os

#servi

cred = credentials.Certificate('/home/pi/PRO/serviceAccount.json')

firebase\_admin.initialize\_app(cred)

db = firestore.client()

def init():

gpio.setmode(gpio.BCM)

gpio.setup(26, gpio.OUT)

gpio.setup(16, gpio.OUT)

gpio.setup(5, gpio.OUT)

gpio.setup(6, gpio.OUT)

gpio.setup(25, gpio.OUT)

def LEDON():

init()

gpio.output(26, True)

print("done LED")

time.sleep(1)

gpio.cleanup()

def LEDOFF():

init()

gpio.output(26, False)

print("done LEDOFF")

time.sleep(1)

gpio.cleanup()

def LEDON2():

init()

gpio.output(25, True)

time.sleep(1)

gpio.cleanup()

def LEDOFF2():

init()

gpio.output(25, False)

time.sleep(1)

gpio.cleanup()

def rotate1():

init()

gpio.output(5, True)

gpio.output(6, False)

sleep(1)

gpio.cleanup()

def rotate2():

init()

gpio.output(5, False)

gpio.output(6, True)

sleep(1)

gpio.cleanup()

#set fire base

def fireseton():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'LIGHT')

doc\_ref.set({

u'led\_s': u'ON'

})

def firesetoff():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'LIGHT')

doc\_ref.set({

u'led\_s': u'OFF'

})

def firesetopen():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'Shutter')

doc\_ref.set({

u'Shutter\_pos': u'open'

})

def firesetclose():

doc\_ref = db.collection(u'SMART-SHELTER').document(u'Shutter')

doc\_ref.set({

u'Shutter\_pos': u'close'

})

#pir

def closed():

o=os.popen(" gnome-terminal -x python3 close.py").read()

print(o)

def pir(sec):

g=0

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(23, gpio.IN)

sleep(4)

i=gpio.input(23)

if i== 0:

if g ==0:#When output from motion sensor is LOW

print("Intruder not detected",i)

sleep(1)

close(1)

g=0

elif i == 1: #When output from motion sensor is HIGH

print ("Intruder detected", i)

g=0

gpio.cleanup()

sleep(2)

if \_\_name\_\_ == '\_\_main\_\_':

try:

print("imported Main")

# Reset by pressing CTRL + C

except KeyboardInterrupt:

print("KeyboardInterrupt by User")

gpio.cleanup()

**ROTATE**

import RPi.GPIO as gpio

import time

from time import sleep

from main import \*

def rotate1():

init()

gpio.output(5, True)

gpio.output(6, False)

sleep(1)

gpio.cleanup()

def rotate2():

init()

gpio.output(5, False)

gpio.output(6, True)

sleep(1)

gpio.cleanup()

#set fire base

def rotate11():

R = 0

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(10, gpio.IN)

i = gpio.input(10)

if i == 1:

print("i=",i)

print("rotated")

break

if i != 1:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(10, gpio.IN)

i= gpio.input(10)

print("i=",i)#When output from gpio is high

gpio.setmode(gpio.BCM)

gpio.setup(5, gpio.OUT)

gpio.setup(6, gpio.OUT)

gpio.output(5, True)

gpio.output(6, False)

gpio.cleanup()

R=1

print("rotaing")

if i == 1:

print("rotated")

break

while True:

gpio.setwarnings(False)

gpio.setmode(gpio.BCM)

gpio.setup(11, gpio.IN)

s = gpio.input(11)

if s != 1:

print("rotated")

rotate1()

print("s=",s)

break

if s == 0 and R == 0:#When output from gpio is low

gpio.setmode(gpio.BCM)

gpio.setup(5, gpio.OUT)

gpio.setup(6, gpio.OUT)

gpio.output(5, False)

gpio.output(6, True)

gpio.cleanup()

print("s=",s)

print("reversing")

if s == 1:

print("rotated")

rotate2()

break

rotate1()

if \_\_name\_\_ == '\_\_main\_\_':

try:

rotate11()

# Reset by pressing CTRL + C

except KeyboardInterrupt:

gpio.cleanup()

print("KeyboardInterrupt by User")

**18.0 ADVANTAGES AND DISADVANTAGES**

**18.1 ADVANTAGES**

* It is more secure.
* Easily accessible.
* Wireless control.

**18.2 DISADVANTAGES**

* It is not portable.
* It needs a power back up.
* It needs high speed internet.
* They need there on space .

**19.0 APPLICATIONS**

Smart shelter can be used for the following applications

* Security for cars
* Parking assistance

**20.0 SCOPE OF IMPROVEMENT**

By adding a cleaning system for car to reduce human effort for the cleaning of the car, it also helps to reduce the wastage of the water

**21.0 CONCLUSION**

Through this project SMART SHELTER we got lots of idea about RASPBERRY PI, motor and its driver, we gain the knowledge about how to design electrical module and mechanical structures.

**22.0 BIBLIOGRAPHY**

**WEBSITE**

* [**www.w3school.com**](http://www.w3school.com)
* [**www.wikipedia.com**](http://www.wikipedia.com)
* [**www.electronicsforu.com**](http://www.electronicsforu.com)
* [**www.raspberrypi.org**](http://www.raspberrypi.org)

**23.0 DATASHEETS**

**24.0 DATASHEET CONTENTS**

|  |  |
| --- | --- |
| **SL. NO** | **CONTENTS** |
| 1 | RASPBERRY PI 3 |
| 2 | GEAR MOTOR |
| 3 | L298D |
| 4 | LM338 |
| 5 | 7805 |
| 6 | PIR SENSOR |
| 7 | WEBCAM |